PNAS

BRIEF REPORT | COMPUTER SCIENCES

OPEN ACCESS

Barriers for the performance of graph neural networks (GNN) in discrete random structures

David Gamarnik[a,1]

Edited by David Weitz, Harvard University, Cambridge, MA; received August 17, 2023; accepted October 7, 2023

Recently, graph neural network (GNN)-based algorithms were proposed to solve a variety of combinatorial optimization problems [M. J. Schuetz, J. K. Brubaker, H. G. Katzgraber,*Nat.Mach.Intell.*4,367–377(2022)].GNNwastestedinparticular on randomly generated instances of these problems. The publication [M. J. Schuetz, J. K. Brubaker, H. G. Katzgraber, *Nat. Mach. Intell.* 4, 367–377 (2022)] stirred a debate whether the GNN-based method was adequately benchmarked against best priormethods.Inparticular,criticalcommentaries[M.C.Angelini,F.Ricci-Tersenghi, *Nat. Mach. Intell.* 5, 29–31 (2023)] and [S. Boettcher, *Nat. Mach. Intell.* 5, 24–25 (2023)] point out that a simple greedy algorithm performs better than the GNN. We do not intend to discuss the merits of arguments and counterarguments in these papers. Rather, in this note, we establish a fundamental limitation for running GNN on random instances considered in these references, for a broad range of choices of GNN architecture. Specifically, these barriers hold when the depth of GNN does not scale with graph size (we note that depth 2 was used in experiments in [M. J. Schuetz, J. K. Brubaker, H. G. Katzgraber, *Nat. Mach. Intell.* 4, 367–377 (2022)]), and importantly, these barriers hold regardless of any other parameters of GNN architecture. These limitations arise from the presence of the overlap gap property (OGP) phase transition, which is a barrier for many algorithms, including importantly local algorithms, of which GNN is an example. At the same time, some algorithms known prior to the introduction of GNN provide best results for these problems up to the OGP phase transition. This leaves very little space for GNN to outperform the known algorithms, and based on this, we side with the conclusions made in [M. C. Angelini, F. Ricci-Tersenghi, *Nat. Mach. Intell.* 5, 29–31 (2023)] and [S. Boettcher, *Nat. Mach. Intell.* 5, 24–25 (2023)].

neural networks | graphs | optimization | algorithms | complexity

A class of problems discussed in ref. 1 and similarly in ref. 2, and attempted to be solved using GNN-based methods falls into the domain of combinatorial optimization in random graphs. For a similar approach regarding the problem of finding ground states of spin glasses, see ref. 3. A graph G is a collection of nodes V and edges E, which is a subset of unordered pairs or, more generally, tuples (hyperedges) of nodes. A generic combinatorial optimization problem is defined by introducing a cost function $C : \{0,1\}^V \to \mathbb{R}$ (also called Hamiltonian in physics jargon), which maps bit strings $\sigma \in \{0,1\}^V$ (aka "decisions") into real values $C(\sigma)$ (aka "cost" or "energy"), and solving the problem $\max_\sigma C(\sigma)$. An equivalent choice of $\sigma \in \{-1,1\}^V$ will be adopted here often for convenience. The presence of various kinds of combinatorial constraints on decisions arising from the presence of edges and hyperedges can be encoded into the cost function C.

A canonical example considered in the aforementioned references is the independent set problem (which we abbreviate as IS) which is an NP-complete in the worst-case problem of finding a largest in cardinality subset $I \subset V$ such that no two nodes are spanned by an edge. Namely, $(i,j) \notin E$ for all $i,j \in I$. This corresponds to a special case of C, where $C(\sigma) = (\sum_{i \in V} \sigma_i) \mathbf{1}(\sigma_i\sigma_j = 0, \forall (i,j) \in E)$. Another example discussed in the same collection of references is the graph maximum cut problem (which we abbreviate as MAXCUT). This is (an NP-complete in the worst-case) problem of partitioning nodes of a graph into two sets which maximizes the number of crossed edges. Formally, this corresponds to the cost function $C : \{-1,1\}^V \to \mathbb{R}$ defined by $C(\sigma) = \sum_{(i,j)\in E} \mathbf{1}(\sigma_i\sigma_j = -1)$. This model extends naturally to hypergraphs as follows. A K-uniform hypergraph is a pair of a node set V and a collection E of hyperedges, where each hyperedge is an unordered subset of K nodes. Thus, 2-uniform hypergraph is just a graph. An extension of MAXCUT to hypergraphs is obtained by considering the cost function $C(\sigma) = \sum_{(i_1,\ldots,i_K)\in E} \mathbf{1}(\sigma(i_1)\sigma(i_2)\cdots\sigma(i_K) = -1)$.

Our last example, arising from the studies of spin glasses, corresponds fixing an order p tensor $J = (J_{i_1,\ldots,i_p}, i_1,\ldots,i_p \in V) \in \mathbb{R}^{n^{\otimes p}}$ and defining $C(\sigma) = \sum_{i_1,\ldots,i_p \in V} J_{i_1,\ldots,i_p}\sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_p}$ for each $\sigma \in \{-1,1\}^V$. The optimization problem is one of finding the value of $\max_\sigma C(\sigma)$.

In the random setting, either the cost function C or the graph G (or both) is generated randomly according to some probability distribution. The setting

Author affiliations: [a]Operations Research Center, Statistics and Data Science Center, Sloan School of Management, Massachusetts Institute of Technology, Cambridge, MA 02140

Author contributions: D.G. designed research; performed research; and wrote the paper.

The author declares no competing interest.

[1]Email: gamarnik@mit.edu.

Published November 6, 2023.

discussed in ref. 1 is IS problem when the underlying graph a random d-regular graph on the set of n nodes denoted for convenience by $V = \{1,...,n\}$. d-regular means every node has exactly d neighbors. The graph is generated uniformly at random from the space of all d-regular graphs on $n$nodes(seerefs.4and5forsomebackgroundregardingexistence and constructions). The random graph constructed this way will be denoted by $\mathbb{G}_d(n)$. The setting of spin glasses corresponds to assuming that the entries of the tensor J are generated randomly and independently from some common distribution with zero mean, such as the standard normal distribution.

Next, we turn to a generic description of GNN algorithms. We follow the notations used in ref. 1. Given a graph $G = (V,E)$, the algorithm generates a sequence of node and time-dependent features $(h_{u,t} \in \mathbb{R}^{d_u}, u \in V, t \geq 0)$. Time is assumed to evolve in discrete steps $t = 0,1,2,...$, and d_u represents the dimension of the feature space for node u. The feature vectors $h_{u,t}$ are generated as follows. The algorithm designer creates a node and time-dependent functions $(f_{u,t}, u \in V, t \geq 0)$ where each $f_{u,t}$ maps $\mathbb{R}^{d_u + \sum_{v \in N(u)}} \to \mathbb{R}^{d_u}$. Here, $N(u)$ denotes the set of neighbors of u (the set of nodes v such that $(u,v) \in E$). The features are then updated according to the rule $h_{u,t+1} = f_{u,t}(h_{u,t}, \{h_{v,t}, v \in N(u)\})$. The update rules $f_{u,t}$ can be parametric or nonparametric (our conclusions do not depend on that) and can be learned using various learning algorithms. The algorithmrunsforacertaintime$t$ $= 0,1,...,R$,whichisalsothe depth of the underlying neural architecture. The obtained vector of features $(h_{u,R}, u \in V)$ is then projected to a desired solution of the problem. As we will see below, the actual details of how the update functions $f_{u,t}$ come about and, furthermore, regardless of the dimensions $d_u, u \in V$ that the algorithm designers opt to work with, the power of GNN algorithms is fundamentally limited by the overlap gap property, which we turn to next.

Limits of GNN

We begin with some background on problems introduced earlier: IS and MAXCUT in a setting of random graphs, and ground states of spin glasses. Let I_n^* denote (any) maximum size independent set in $\mathbb{G}_d(n)$, which we recall is a random d-regular graph, and $|I_n^*|$ denote its size (cardinality). The following two facts were established in refs. 6 and 7 respectively. For each d, there exists α_d such that $|I_n|/n$ converges to α with high probabilityas$n \to \infty$.Furthermore,$\alpha_d = 2(1+o_d(1))\log d/d$. Here, $o_d(1)$ denotes a function which converges to zero as $d \to \infty$. Informally, we summarize this by saying that the size $|I_n^*|$ of a largest independent set in $\mathbb{G}_d(n)$ is approximately $2(\log d/d)n$.

Next, we turn to the discussion of algorithms for finding large independent sets in $\mathbb{G}_d(n)$. It turns out that the best-known algorithm for this problem is in fact the Greedy algorithm (the algorithms discussed in refs. 8 and 9) which recovers a factor 1/2-optimum independent set. More precisely, let I_{Greedy} be the independent set produced by the Greedy algorithm for

$\mathbb{G}_d(n)$.Then,$\lim_{d\to\infty} \alpha_d^{-1} \lim_n (|I_{\text{Greedy}}|/n) = 1/2asd \to \infty$, Exercise 6.7.20 in ref. 5. No algorithm is known which beats Greedy by a factor nonvanishing in d.

The theory based on the overlap gap property (OGP) explains this phenomenon rigorously. The OGP for this problem was established in ref. 10 and it reads as follows:

For every factor $1/2 + 1/(2\sqrt{2}) < \theta < 1$, there exists $0 < \nu_1 < \nu_2 < 1$ such that for every two independent sets I_1, I_2 which are θ-optimal, namely $|I_1|/n \geq \theta\alpha_d$, $|I_2|/n \geq \theta\alpha_d$, it is the case that either $|I_1 \cap I_2|/n \leq \nu_1$ or $|I_1 \cap I_2|/n \geq \nu_2$, f or all large enough d, with high probability as $n \to \infty$. Informally, every two sufficiently large independent sets (namely those which are multiplicative factor θ-close to optimality) are either "close" to each other (overlap in at least $\nu_2 n$ many nodes) or "far" from each other (overlap in at most $\nu_1 n$ many nodes). Namely, solutions to the IS optimization problem with sufficiently large optimization values exhibit a gap in the overlaps (hence the name of the property).

It turns out that OGP is a barrier to a broad class of algorithms, in particular, algorithms which are local in an appropriately defined sense. This was established in the same paper (10). We introduce the notion of locality only informally. The formal definition involves the concept of Factors of IID for which we refer the reader to ref. 10. An algorithm, which maps graphs G to an independent set in G is called R-local if for every node u of the graph G, the algorithmic decision as to whether to make this node a part of the constructed independent set or not is based entirely on the size R neighborhood of this node u. In particular, we see that the GNN algorithm is R-local provided that the number of iterations t of GNN is at most R. Importantly, this holds regardless of the complexity of the feature dimensions d_u and the choice of update functions $f_{u,t}$. We recall that the GNN algorithm reported in ref. 1 was based on 2 iterations and as such it is 2-local, that is, $R = 2$.

A main theorem proved in ref. 10 states that OGP is a barrier for all R-local algorithms, as long as R is any constant not growing with the size of the graph. Specifically, for any R, consider any algorithm $\mathbb{A}$ which is R-local. Then, the independent set produced by $\mathbb{A}$ is at most $(1/2 + 1/(2\sqrt{2}))\alpha_d$ for large enough d with high probability as $n \to \infty$. Using a more sophisticated notion of multioverlaps, the result was improved in ref. 11 to factor 1/2 of optimality for the same class of all local algorithms. Importantly, as we recall, 1/2 is the threshold achievable by the Greedy algorithm. The result was recently extended to the class of algorithms based on low-degree polynomials and smalldepth Boolean circuits in refs. 12 and 13. It is conjectured that beating the 1/2 threshold is not possible within the class of polynomial time algorithms (but showing this will amount to proving $P \neq NP$).

As a consequence of the discussion above, we obtain an important conclusion regarding the limits of GNN for solving the IS problem in $\mathbb{G}_d(n)$.

Theorem 0.1. *Consider any architecture of the GNN algorithm with any choice of dimensions $(d_v, v \in \{1,2,...,n\})$, any choice of feature functions $h_{u,t}$, and any choice of update functions $f_{u,t}$. Suppose the GNN algorithm iterates for R steps and produces an independent set I_{GNN} in the random regular graph $\mathbb{G}_d(n)$. Then, the size of I_{GNN} is at most half-optimum asymptotically in d, for any value of R.*

We stress here that the depth parameter R can be arbitrarily large and, in particular, may depend on the average degree d, provided it does not depend on the size n of the graph. We recall that $R = 2$ in the implementation reported in ref. 1. Since the Greedy algorithm already achieves $1/2$ optimality, as we have remarked earlier, this result leaves very little space for the GNN to outperform the known (Greedy) algorithm for the IS defined on random regular graphs.

Next, we turn to the MAXCUT problem on random graphs and random hypergraphs. The situation here is rather similar, but better developed in the context of random Erdös-Rényi graphs and hypergraphs, as opposed to random regular graphs, and thus, this is the class of random graphs we now turn to. A random Erdös-Rényi graph with average degree d denoted by $\mathbb{G}(n,d)$ is obtained by connecting every pair of nodes i,j among n nodes with probability d/n, independently across all unordered pairs $i \neq j$. A random K-uniform hypergraph is obtained similarly by creating a hyperedge from a collection of nodes $i_1,\ldots,i_K$ with probability $d/\binom{n-1}{K-1}$. We denote this graph by $\mathbb{G}(n,d;K)$ It is easy to see that the average degree in both $\mathbb{G}(n,d)$ and $\mathbb{G}(n,d;K)$ is $d + o(1)$. It was known for a while that the optimum values of MAXCUT in $\mathbb{G}(n,d;K)$ are of the form $n(d/(2K) + \gamma_K^* \sqrt{d} + o(\sqrt{d}))$ as $n \to \infty$, (14), for some constant γ_K^*. Namely, the optimum value is known up to the order $n\sqrt{d}$. γ_K^* was computed in refs. 15 and 16 first for the case $K = 2$ and then extended to general K in ref. 17. As it turns out, γ_K^* is the value of the ground state of a K-spin model, known since the work of Parisi (18), Talagrand (19), and Panchenko (20).

Interestingly, as far as the algorithms are concerned, there is a fundamental difference between the case $K = 2$ (aka graphs) versus $K \geq 3$. Specifically, algorithms achieving the asymptotically optimal value $n(d/(2K) + \gamma_K^* \sqrt{d} + o(\sqrt{d}))$ are known based on local Approximate Message Passing (AMP) schemes (21). Furthermore, conjecturally, the OGP does not hold for this problem. However, when $K \geq 4$ and is even, OGP provably does hold and again presents a barrier to all local algorithms, as was established in ref. 17. Furthermore, a sophisticated version of the multi-OGP called BranchingOGP was computed (22), the threshold for which, denoted by $\gamma_{\text{B-OGP},K}$ matches the best-known algorithms, which is again the AMP type. The formal statement of the OGP is very similar to the one for the IS and we skip it. As an implication, we obtain our second conclusion.

Theorem 0.2. *Consider any architecture of the GNN algorithm which produces a partition* $\sigma_{\text{GNN}} \in \{\pm 1\}^n$ *in the random hypergraph* $\mathbb{G}(n,d;K)$. *Suppose* $K \geq 4$ *and is even. For sufficiently large degree values d, the size of the cut associated with this solution is at most* $n(d/(2K) + \gamma_{\text{B-OGP},K}\sqrt{d} + o(\sqrt{d}))$ *with high probability, for any choice of* R. *This is suboptimal since* $\gamma_{\text{B-OGP},K} < \gamma_K^*$.

As the threshold $\gamma_{\text{B-OGP},K}$ is achievable by the AMP algorithm, again this leaves very little space for GNN to outperform the bestknown (namely AMP) algorithm for this problem when $K \geq 4$. We should note though that when $K = 2$ the optimality might be achievable within the framework of GNN, as in this case, the optimality is reachable by local (AMP) algorithms. In any event though, GNN will not perform stronger than any local algorithm.

The story for the problem of finding near ground states in spin glasses is very similar and is skipped. We refer the reader to surveys (23) and (24) for details.

Discussion

In this paper, we have presented barriers faced by GNN-based algorithms in solving combinatorial optimization problems in random graphs and random structures. These barriers stem from the complex solution space geometry property in the form of the OGP, a known barrier to broad classes of algorithms, local algorithms in particular. As GNN falls within the framework of local algorithms, OGP is a barrier to GNN as well. Since algorithms are known which achieve the optimality values just below the OGP phase transition threshold, this leaves very little room for the GNN to outperform the known algorithms.

Data, Materials, and Software Availability. There are no data underlying this work.

ACKNOWLEDGMENTS. Funding for this work was provided by National Science Foundation grant 2233897.

1. M. J. Schuetz, J. K. Brubaker, H. G. Katzgraber, Combinatorial optimization with physics-inspired graph neural networks. *Nat. Mach. Intell.* 4, 367–377 (2022).
2. M. J. Schuetz, J. K. Brubaker, Z. Zhu, H. G. Katzgraber, Graph coloring with physics-inspired graph neural networks. *Phys. Rev. Res.* 4, 043131 (2022).
3. M. Shen *et al.*, "Finding spin glass ground states through deep reinforcement learning" in *APS March Meeting Abstracts*, (2022), vol. 2022, pp. K09-001.
4. S. Janson, T. Luczak, A. Rucinski, *Random Graphs* (John Wiley & Sons, 2011), vol. 45.
5. A. Frieze, M. Karonski,´ *Introduction to Random Graphs* (Cambridge University Press, 2015).
6. M. Bayati, D. Gamarnik, P. Tetali, Combinatorial approach to the interpolation method and scaling limits in sparse random graphs. Ann. Probab. 41, 4080–4115 (2013). Conference version in Proc. 42nd Ann. Sym. Theory Comput. (STOC) (2010).
7. A. Frieze, T. Łuczak, On the independence and chromatic numbers of random regular graphs. *J. Combinat. Theory Ser. B* 54, 123–132 (1992).
8. M. C. Angelini, F. Ricci-Tersenghi, Modern graph neural networks do worse than classical greedy algorithms in solving combinatorial optimization problems like maximum independent set. *Nat. Mach. Intell.* 5, 29–31 (2023).
9. S. Boettcher, Inability of a graph neural network heuristic to outperform greedy algorithms in solving combinatorial optimization problems. *Nat. Mach. Intell.* 5, 24–25 (2023).
10. D. Gamarnik, M. Sudan, Limits of local algorithms over sparse random graphs. *Ann. Probab.* 45, 2353–2376 (2017).
11. M. Rahman, B. Virag, Local algorithms for independent sets are half-optimal. *Ann. Probab.* 45, 1543–1577 (2017).
12. D. Gamarnik, A. Jagannath, A. S. Wein, "Low-degree hardness of random optimization problems" in *61st Annual Symposium on Foundations of Computer Science*, (IEEE, New York, 2020).
13. A. S. Wein, Optimal low-degree hardness of maximum independent set. *Math. Stat. Learn.* 4, 221–251 (2022).
14. D. Coppersmith, D. Gamarnik, M. Hajiaghayi, G. Sorkin, Random MAXSAT, random MAXCUT, and their phase transitions. *Random Struct. Algorithms* 24, 502–545 (2004).
15. A. Dembo, A. Montanari, S. Sen, Extremal cuts of sparse random graphs. *Ann. Probab.* 45, 1190–1217 (2017).
16. S. Sen, Optimization on sparse random hypergraphs and spin glasses. *Random Struct. Algorithms* 53, 504–536 (2018).
17. W. K. Chen, D. Gamarnik, D. Panchenko, M. Rahman, Suboptimality of local algorithms for a class of max-cut problems. *Ann. Probab.* 47, 1587–1618 (2019).
18. G. Parisi, A sequence of approximated solutions to the SK model for spin glasses. *J. Phys. A: Math. General* 13, L115 (1980).
19. M. Talagrand, The parisi formula. *Ann. Math.* 231, 221–263 (2006).
20. D. Panchenko, *The Sherrington-Kirkpatrick Model* (Springer Science & Business Media, 2013).
21. A. E. Alaoui, A. Montanari, M. Sellke, Local algorithms for maximum cut and minimum bisection on locally treelike regular graphs of large degree. *Random Struct. Algorithms* 63, 689–715 (2023).

22. B. Huang, M. Sellke, "Tight lipschitz hardness for optimizing mean field spin glasses" in *IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE, 2022), pp. 312–322.
23. D. Gamarnik, The overlap gap property: A topological barrier to optimizing over random structures. *Proc. Natl. Acad. Sci. U.S.A.* 118, e2108492118 (2021).
24. D. Gamarnik, C. Moore, L. Zdeborová, Disordered systems insights on computational hardness. *J. Stat. Mech.: Theory Exp.* 2022, 114015 (2022).